

# Distance entre deux points [bs10] - Exercice

Karine Zampieri, Stéphane Rivière

Unisciel  algoprog  Version 13 mai 2018

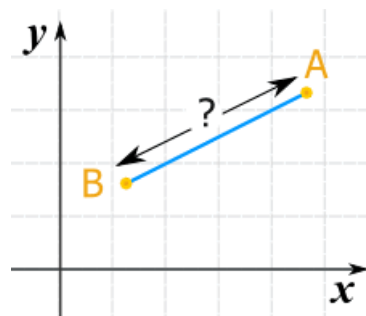
## Table des matières

<b>1</b>	<b>Distance entre deux points / pgdistance</b>	<b>2</b>
1.1	Calcul de la distance . . . . .	2
1.2	Calcul de la cote . . . . .	2
1.3	Validation . . . . .	3
<b>2</b>	<b>Références générales</b>	<b>3</b>

## Python - Distance entre deux points (Solution)

 Mots-Clés Structures de base ■  
Difficulté ●○○

 **Objectif**  
Cet exercice calcule la distance entre deux points du plan ainsi que la cote de chacun des points dans l'espace. (image : <https://www.mathsisfun.com>)



...(énoncé page suivante)...

# 1 Distance entre deux points / pgdistance

## 1.1 Calcul de la distance



Écrivez un script qui saisit les coordonnées de deux points du plan  $(x_1, y_1)$  dans `x1`, `y1` pour le premier et  $(x_2, y_2)$  dans `x2`, `y2` pour le deuxième. Toutes les données sont réelles. Affichez les invites :

```
Premier point?
Deuxième point?
```



Calculez la distance entre les deux points définie par :

$$distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

### Outil Python

La fonction `sqrt(x)` de  $\sqrt{x}$  est définie dans la bibliothèque `math`.



Affichez (où `[x]` désigne le contenu de `x`) :

```
La distance est [...]
```



Testez. Exemple d'exécution :

```
Premier point? -4 3
Deuxieme point? 3.1 4.1
La distance est 7.18471
```

## 1.2 Calcul de la cote

Étant donné les coordonnées  $x$  et  $y$  d'un point de l'espace, sa cote est définie par :

$$z = (x^2 + y^2) \exp\left(-\sqrt{x^2 + y^2}\right) \cos\left(\sqrt{x^2 + y^2}\right)$$



Calculez la cote de chacun des points dans `z1` et `z2` respectivement.

### Orientation

L'expression à calculer contient plusieurs sous-expressions qu'il vaut mieux ne pas recalculer, autant pour accélérer le fonctionnement de l'algorithme que pour simplifier son écriture.

**Outil Python**

Les fonctions `cos(x)` du cosinus et `exp(x)` de l'exponentielle sont également définies dans la bibliothèque `math`.



Affichez la cote de chacun des points sous la forme suivante :

Cote de (`[x1]`,`[y1]`) est `[z1]`



Testez. Exemple d'exécution :

```
Premier point? -4 3
Deuxieme point? 3.1 4.1
La distance est 7.18471
Cote de (-4,3) est 0.0477825
Cote de (3.1,4.1) est 0.0641817
```

### 1.3 Validation



Validez votre script avec la solution.

**Solution Python** @[pgdistance1.py]

```
import math

def PGDistance1():
    print("Premier point? ", sep="", end="")
    x1 = float(input())
    y1 = float(input())
    print("Deuxieme point? ", sep="", end="")
    x2 = float(input())
    y2 = float(input())
    dist = math.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1))
    print("La distance est ", dist, sep="")
    sc1 = x1 * x1 + y1 * y1
    module1 = math.sqrt(sc1)
    z1 = sc1 * math.exp(-module1) * math.cos(module1)
    print("Cote de (", x1, ", ", y1, ") est ", z1, sep="")
    sc2 = x2 * x2 + y2 * y2
    module2 = math.sqrt(sc2)
    z2 = sc2 * math.exp(-module2) * math.cos(module2)
    print("Cote de (", x2, ", ", y2, ") est ", z2, sep="")

PGDistance1()
```

## 2 Références générales

Comprend [Maysonnave-AL1 :c2] ■